Using Small XML Elements to Support Relevance

Georgina Ramírez, Thijs Westerveld and Arjen P. de Vries
CWI, Amsterdam, The Netherlands
georgina@cwi.nl, thijs@cwi.nl, arjen@acm.org

ABSTRACT

Small XML elements are often estimated relevant by the retrieval model but they are not desirable retrieval units. This paper presents a generic model that exploits the information obtained from small elements. We identify relationships between small and relevant elements and use this *linking* information to reinforce the relevance of other elements before removing the small ones. Our experiments using the INEX testbed show the effectiveness of our approach.

Categories and Subject Descriptors: H.3.3 [Information Storage and Retrieval]: Information Search and Retrieval.

General Terms: Algorithms, Experimentation.

Keywords: XML retrieval, score propagation, length normalization.

1. INTRODUCTION

In an XML retrieval setting each of the elements of an XML document, ranging from elements with only few terms (such as titles) to elements with thousands of terms (such as articles) could be presented to the user as a result.

However, since small elements like figure captions or titles contain insufficient information to answer an information need on their own, it is common belief that retrieval systems can safely remove these small elements from their candidate lists. A range of techniques have been developed to effectively remove the *too* small elements from the result set. Examples include the removal of small elements from the index (or filtering them from the results list) [1], the prior definition of a subset of retrievable XML element types [5], or the introduction of a length prior to reward larger elements and punish shorter ones [3]. These techniques improve the ranking because the larger elements get ranked higher. However, when removing the small elements, the evidence collected by the retrieval model about the relevancy of these is ignored.

This poster investigates how the small elements may function as indicators of relevance in the document, helping to identify and rank higher their related, larger elements.

SIGIR'06, August 6–11, 2006, Seattle, Washington, USA.
ACM 1-59593-369-7/06/0008.

2. LINKS BETWEEN SMALL AND RELEVANT ELEMENTS

To learn how small elements with a high similarity to the query relate to relevant elements, we analyse the difference between retrieved elements in a baseline run and relevant elements as identified in the INEX assessments [2]. Our analysis is based on the top 1000 retrieved elements using a simple language modeling approach, which treats each element as a separate 'document' (see [4]). We study the occurrence of relevant elements in the direct vicinity of each retrieved small element. We only report on the relationships found in the body part of the articles. Similar relationships can be found in the frontmatter and backmatter.

We created an histogram of the number of levels in the XML tree that we have to go up from a retrieved small element to reach a relevant ancestor. Retrieved `st` elements (section titles) are rarely relevant themselves, but their containing element, one level up, often is. The same holds for `it` (italics) and `fig` (figure) elements. Retrieved `fgc` (figure captions) tend to have a relevant grandparent. Retrieved small paragraphs (`p` and `ip1`) are mostly relevant themselves. These statistics motivate two types of links:

Support links Those links where the small element types are rarely relevant themselves but some ancestor is. These elements can therefore *support* or re-inforce the relevance of other elements, but they should not be retrieved themselves in any case. Examples include section titles and italics.

Propagation links Those links where the small element type is mostly relevant itself. These links could be retrieved themselves if they would not be too small. Since they do contain relevant information, a reasonable approach is to reward the (larger) elements that contain them. Thus, we link them to the first *sensible* retrieval unit found up the tree, i.e., a unit with at least 30 words. Examples of *propagation* links found in the INEX collection are paragraphs (p and ip1) and list items (item).

Figure 1 shows the discovered relations on an XML tree.

Before removing small elements from our results, we use the linked relationships to propagate their score. The final retrieval status value (RSV) of an element E_j is computed as:

$$RSV(E_j) = f(P_{RM}(E_j), aggr_{i \in inlinks(E_j)}(P_{RM}(i))) \quad (1)$$

Where $P_{RM}(\cdot)$ is the score given to an element by the retrieval model; $aggr$ is an aggregation function that combines

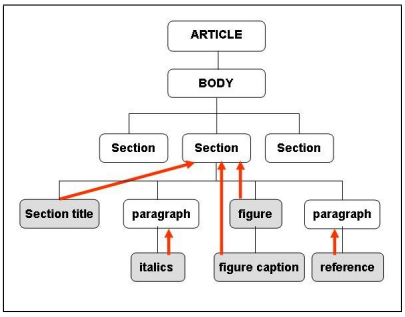

Figure 1: Subset of article structure with added links.

the scores of all the elements that point to E_j, and $f(x, y)$ is a function to combine the score of the elements and the aggregated score of their inlinks.

3. RESULTS

The main goal of our experiments is to analyse the performance, in terms of effectiveness, of the individual and global contribution of using different number and types of links.

Our baseline is a simple language modeling approach, which treats each element as a separate 'document' (see [4]). We used $\lambda = 0.5$ and a cut-off value of 30 (elements containing less than 30 words are removed).

We use the *max* operator as aggregation function in Equation 1; the intuition behind this decision is that if an element has one small relevant element pointing to it, it is enough to be rewarded and no more evidence is needed. We study the performance of two different operators when combining the score obtained from the inlinks and the original score of the XML elements. We think that this is a more problematic combination because we want to reward the elements for containing relevant support links but we do not want to overrule too much the relevance score of the retrieval model. We experiment with the *max* and the *average* functions.

Table 1 shows the results obtained[1]. The first two rows show the results of using only the *support* links versus using both types of links. We can see that for very high precision (nxCG[10]) our approach performs significantly better when using the *support* links. This improvement is even better when using the max operator instead of the average, which means that small elements are indeed a good pointer to relevance information. Clearly, the re-ranking produced by the small element scores before being removed is very positive for retrieval effectiveness. There is no significant improvement when using the *support* links in combination with the *propagation* links. One reason for that can be that paragraphs (e.g.) can get a high score for simply containing a single query term multiple times. This might lead to reward the wrong sections. This effect does not occur with the *support* link elements (e.g. section titles), since they usually do not contain duplicated terms; when they have a high score it is because they contain all or most of the query terms. Looking at the recall-oriented measure (Maep), we see that our approach does not hurt recall. Additional experiments show that recall can even be improved when combined with techniques such as article weighting [4]. Regarding the contribution of different types of small elements, we clearly see that the section titles contribute significantly to improve effectiveness on high precision levels but also slightly on recall. It is reasonable to agree that section titles can be very good pointers to relevant information (e.g., sections). On the other hand, types such as italics do not perform that well comparably. This could simply be due to the (comparably) small number of elements we retrieve from these types. The fact that they do not hurt performance when combined with the section titles can be a good indicator that they are also good pointers. More experiments are needed to confirm this hypothesis. When analysing the performance of the other type of small elements, we do not observe significant improvements. Again, this might be due to the small amount of elements retrieved for each type.

[1]Information about the metrics used can be found in [2]. The plus symbols indicate a significant increase over the baseline using the Wilcoxon signed-rank test at a confidence level of 95% (+) or 99% (++).

Table 1: Usage of link information. Support Links (SL) and Support+Propagation Links (SPL). Results using AVG (MAX) as combination function.

	nxCG[10]	nxCG[25]	Maep
base	0.246	0.254	0.076
SL	0.283+ (0.301++)	0.260 (0.264+)	0.077 (0.078)
SPL	0.270 (0.266)	0.2515 (0.251)	0.074 (0.074)
it	0.250 (0.253)	0.256 (0.253)	0.076 (0.077)
st	0.287++ (0.291++)	0.261+ (0.265+)	0.077 (0.079+)
it+st	0.285+ (0.301++)	0.260 (0.261+)	0.077 (0.078)

4. CONCLUSIONS

Traditional approaches based on size or element type are useful, but they ignore the important contextual clues that the small elements can bring. We have shown that adding explicit links from small elements to other elements helps in locating relevant elements on the tree. Mainly the precision at high recall levels is significantly improved.

The link detection and analysis techniques presented in this paper have been shown to be effective in other retrieval tasks (such as focussed retrieval) [6] and in combination with other XML retrieval techniques. They can also easily be applied to longer elements. For example, abstracts may be good supporting elements for articles. Another interesting direction for future research is to exploit the discovered link structure in relevance feedback or query expansion.

5. REFERENCES

[1] C. L. A. Clarke. Controlling overlap in content-oriented XML retrieval. In *SIGIR 2005*, pages 314–321. ACM Press, 2005.

[2] N. Fuhr, M. Lalmas, S. Malik, and G. Kazai, editors. *Inititive for the evaluation of XML Retrieval. Proceedings of the Fourth Workshop (INEX 2005)*, Germany, 2005.

[3] J. Kamps, M. de Rijke, and B. Sigurbjörnsson. Length Normalization in XML Retrieval. In *SIGIR 2004*, pages 80–87. ACM Press, 2004.

[4] J. List, V. Mihajlović, G. Ramírez, A. P. de Vries, D. Hiemstra, and H. E. Blok. TIJAH: Embracing IR Methods in XML Databases. *Information Retrieval journal*, 8(4):547–570, December 2005.

[5] Y. Mass and M. Mandekbrod. Retrieving the most relevant XML Components. In N. Fuhr, S. Malik, and M. Lalmas, editors, *INEX 2003 Workshop Proceedings*, 2003.

[6] G. Ramírez, T. Westerveld, and A. de Vries. Using Structural Relationships for Focused XML Retrieval. In *FQAS 2006*. Springer, 2006.